EngOpt 2012 – 3rd International Conference on Engineering Optimization
Rio de Janeiro, Brazil, 01 - 05 July 2012.

Constraints Handling for Hybrid Algorithms in Waterflooding Optimization Problem

Leonardo Correia de Oliveira[1], Silvana Maria Bastos Afonso[1], Bernardo Horowitz[1] and Afonso Celso de Castro Lemonge[2]

[1]Universidade Federal de Pernambuco – Dept. de Engenharia Civil – Estruturas
Av. Acadêmico Hélio Ramos, S/N, Recife – PE – CEP 50740-530, Brasil
[2]Universidade Federal de Juiz de Fora – Faculdade de Engenharia – Dept. de Mecânica Aplicada e Computacional
Campus Universitário, Martelos, S/N, Juiz de Fora – MG – CEP 36036-330, Brasil
Email: leonardo.oliveira@ufpe.br, smb@ufpe.br, horowitz@ufpe.br, afonso.lemonge@ufjf.edu.br

Abstract

In oil Reservoir Engineering applications, a problem of great interest is the dynamic optimization of waterflooding management. In this work the net present value (NPV) is taken as the function to be maximized. The design variables are the allocated rates at producers and injectors wells. Usually the concession period is subdivided into a number of control cycles, whose switching times are fixed and the well rates are applied. Alternatively, the switching times can be considered as design variables in the optimization formulation problem. This assumption increases flexibility.

In management and can decreases the total number of variables for similar recovery efficiency. In despite of this, the formulation of this problem leads to a highly nonlinear, multimodal objective function. An adequate choice to solve this problem is to consider a global optimization strategy. Due to the cost of the reservoir analysis and the convergence behavior for this kind of algorithms, a hybrid optimization strategy is proposed considering surrogate models. The hybrid strategy combines different methods at different stages of the process in order to exploit the best features of each methodology. The focus is to balance a global search process with the precision and efficiency of a local search procedure. Here, the global search is driven by the genetic algorithm (GA) and the local search is driven by the sequential approximation optimization (SAO) method. As such algorithm is suitable to solve unconstrained problems, techniques to handle the constraints of the waterflooding problem are proposed in this work. Different strategies may be used to include general constraints. Among them it is considered an adaptive penalty method which does not require any type of user defined penalty parameter and uses information from the population. Moreover, to make the global search procedure more effective a filter scheme is proposed in which the level of feasibility of the initial population is assured.

Keywords: production optimization, petroleum reservoir, hybrid strategy, constraints handling

1. Introduction

In reservoir engineering a problem of great interest is the management of the field [1]. Reservoir simulation is used extensively to identify opportunities to increase oil production in heavy oil reservoirs. In this scenario waterflooding is the most common method used to improve oil recovery after primary depletion. The management of the field can be formulated as an optimization problem in which the rates in the producers and injection wells are obtained fulfilling specific constraints.

From a point of view of economics, profitability is usually chosen as the objective function of the problem, where the maximization of the net present value can be seek, for example. In the problem formulation, usually the concession period is divided into control cycles, whose flow rates from each well are exchanged in fixed times during the period. The time when the exchanges are applied can also be considered as design variables. Under such consideration, the flexibility in the management is greatly increased. This assumption, however provides a mathematical formulation that leads to an objective function with multimodal characteristics.

In general, the numerical simulation of above application has high computational cost, even considering the hybrid strategy. Therefore the multiple numerical simulations required in the optimization procedure are prohibitively expensive. In this context, approximation strategies appears as a powerful tool to overcome the abovementioned problem. Here, the simplified model will be used to provide insight on the overall trends of the objective and constraint functions over the investigated design space. In this work we will consider kriging data fitting approximation approach.

The optimization problem described above would be adequate to be solved considering a global optimization strategy. As natural choices for this type of solution emerges the class of evolutionary algorithms such as genetic algorithms, swarms methodologies among others [24]. In general such algorithms, present fast convergence at initial stages of global search, but in the neighborhood of the global optimum, the search process becomes very slow. To overcome that, mathematic programming algorithms would be ideal to be combined with global strategies as they provide very fast convergence around a pre specified initial point.

The resulting combined tool, the hybrid strategy, will provide a balance between a global search process and the efficiency of a local process. In this sense, such strategy appears as a promising scheme to solve the dynamic optimization of waterflooding management problem, as a computation reduction cost can be obtained without comprising the global search properties. As global strategy the genetic algorithm (GA) [7] is used and is combined to a local strategy that uses a sequential approximation optimization

(SAO) [15, 14].

The surrogate model is applied in a different way on each strategy considered during the optimization process. When the global search is executed, the built surrogate model covers the whole domain, which might be updated during the search. When the local search is executed, the surrogate model is applied on a iterative process, which decomposes the original problem into a sequence of optimization subproblems, confined on small subregions of the domain, called trust regions.

In general, evolutionary algorithms are suitable to solve unconstrained problems, but there are some techniques which make it suitable to handle the constraints. To handle the constraints here, it was considered an adaptive penalty method which does not require any type of user defined penalty parameter and uses information from the population. Such implementation is compared with results obtained using an ordinary penalty method found in MATLAB global optimization toolbox. Moreover, to make the global search procedure more effective a filter scheme is proposed in which the level of feasibility of the initial population is assured.

2. Waterflooding Problem

The waterflooding problem is by far the most commonly used method to improve oil recovery after primary depletion. In spite its many favourable characteristics, reservoir heterogeneity; permeability contrast in particular, can adversely impact the performance of water flooding. Moreover, it is well known that the presence of high permeability streaks can severely reduce the sweep efficiency leading to an early water arrival at the producers and by-passed oil. One approach to counteract the impact of the heterogeneity and to improve water flood sweep efficiency is through optimal rate allocation to the injectors and producers.

The time when the control cycles are exchanged can be considered as design variables, increasing the flexibility of the problem. As a consequence, the total number of variables can be increased, but this consideration would lead to a smarter optimal water flood sweeping, the necessary pressure on reservoir, and the shutting off some wells. Due to this additional flexibility, it is possible to obtain an optimal management adopting less control cycles.

The dynamic optimization of procedure scheduling, considering the NPV as the objective function and constraints at platform's total rate is formulated as follows:

$$\begin{aligned}
&\text{Maximize: } \text{VPL} = f(x_{p,t}, x_{\Delta t,k}, \boldsymbol{u}) = \sum_{t=0}^{T}\left[\frac{1}{(1+d)^t}\cdot F_t(x_{p,t}, x_{\Delta t,k}, \boldsymbol{u})\right] \\
&\text{subject to: } \sum_{p\in P} x_{p,t} \le 1, \quad t = 1 \ldots n_t \\
&\sum_{p\in I} x_{p,t} \le 1, \quad t = 1 \ldots n_t \\
&\sum_{p\in P} x_{p,t} \le \sum_{p\in I} \frac{x_{p,t}}{\alpha} \le \frac{1}{\alpha}\cdot \sum_{p\in P} x_{p,t}, \quad t = 1 \ldots n_t \\
&\sum_{k=1}^{n_t-1} x_{\Delta t,k} \le 1 \\
&x_{p,t}^{l} \le x_{p,t} \le x_{p,t}^{u} \\
&x_{\Delta t,k}^{l} \le x_{\Delta t,k} \le x_{\Delta t,k}^{u}, \quad k = 1 \ldots n_t - 1
\end{aligned} \tag{1}$$

In above equations, d is the discount rate applied on capital, T is the concession period, and F_t is the cash flow at time t. The term P and I refer to production wells and injection wells, respectively, the $\boldsymbol{u}$ vector represents parameters that cannot be controlled, as rocks and fluids' properties on reservoir and, at least, there are $x_{p,t}$ and $x_{\Delta t,k}$ as design variables. The $x_{p,t}$ variable is the allocated rate for well p at time t, given by:

$$x_{p,t} = \frac{q_{p,t}}{Q_{prod,\max}},\ p \in P\ ; \quad x_{p,t} = \alpha\frac{q_{p,t}}{Q_{prod,\max}},\ p \in I \tag{2}$$

where $q_{p,t}$ is the well rate, $Q_{prod,\max}$ is the maximum allowed total fluid rate o production wells and α is the ratio determined between the $Q_{prod,\max}$ and the maximum allowed total water injected, represented by $Q_{inj,\max}$.

The $x_{\Delta t,k}$ variable represents the time variables, where the exchanges on wells' rates will be applied, and it is defined in normalized due to T, represented by:

$$x_{\Delta t,k} = \frac{\Delta t_k}{T},\ k = 1 \ldots n_t - 1 \tag{3}$$

3. Approximation

In the process of surrogate model construction the challenge is to provide a substitute model with sufficient accuracy. Kriging approximations (in local or in global context) [19] will be considered here. The central idea of this scheme is that the sample response values exhibit spatial correlation with response values modeled by a Gaussian process around each sample location. The main advantages pointed for such scheme are: the ability to accommodate irregularly space data, the ability to model functions with many peaks and valleys together with the exact interpolation of the given sample response.

As this is a data fitting based model, the first step is to generate the sampling points. A design of experiments (DoE) approach determines the points in the design space. To choose a good sample is important to obtain a good surrogate model because the model created is strongly influenced by the point location. The reason for this is that each point from the sample is evaluated in the real function and the results are used to create the surrogate model. The appropriate sample considers a minimum number of points but ensures accuracy.

The samples can be obtained by a variety of available methods, including Latin Hypercube Sampling (LHS), Orthogonal Array (OA), Centroidal Voronoi Tesselltion (CVT), Quasi Monte Carlo (QMC), Simple Ramdom Sample (SRS) [2].

For these methods, one measure of a point set's uniformity of precision onto all the coordinates axes is called discrepancy. As projection uniformity increases, discrepancy decreases. LHS is a lower-discrepancy sampling method than CVT is. Methods specifically designed with low discrepancy in mind are the quasi- or sub-random low-discrepancy sequence methods [21]. Though CVT tends to have better volumetric uniformity than the sequence methods, which helps its relative performance in other areas, it also has much higher discrepancy, which deteriorates its relative performance. Therefore, a hybrid of CVT and LHS has recently been formulated [8] which appears to have both lower discrepancy than pure CVT and higher volumetric uniformity than pure LHS. The method is called "Latinized" CVT (LCVT) [20]. The LCVT method is considered to provide the samples on this work.

The kriging fitting scheme is the next step used to develop the predictor and error expressions in order to evaluate the functions at untried design points. Details of used implementation can be seen elsewhere [17, 4].

4. Optimization Strategies

To tackle the solution of the problem previously formulated, global and local optimization strategies will be combined with a numerical reservoir engineering simulator in a unique computation tool. The output of interest is high nonlinear, with multimodal characteristics. Also depending on the well constraints settings, the NPV response can be discontinuous as indicated in reference [17].

Algorithms suited for this such as GAs [7] are often expensive since they usually require many function evaluations and are very slow to reach convergence. Thus, these algorithms should only be used to identify promising optimum areas in the design space. Once such regions have been found, a local scheme can be used to converge on precise optimum. This is the hybrid approach which will be here built combining the GA and SAO:

4.1. Global Strategy

As previously mentioned, GA will be the global strategy considered on optimization process. GA's, as powerful and broadly applicable stochastic search and optimization techniques, are perhaps the most widely known types of evolutionary computation methods today. In the past decade the GA community has turned much of attention to optimization problems in industrial engineering resulting in a fresh body of research and applications [12].

In general, a genetic algorithm has five basic components, as summarized by Michalewicz [23]:

1. A genetic representation of solutions to the problem;
2. A way to create an initial population of solutions;
3. An evolution function rating solutions in terms of their fitness;
4. Genetic operators that alter the genetic composition of children during reproduction;
5. Values for the parameters of GA.

In general, GA's follow the steps below:

1. Generation of initial population;
2. Evaluate the fitness of all individuals in population;
3. Check for convergence;
4. Evaluation of reliability of each chromosome;
5. Application of genetic operators to generate new population (selection, crossover and mutation);
6. Return to step 2 and repeat the process until the convergence is reached.

The GA adopted in this work was the one found in global optimization toolbox from MATLAB [18].

It is well known that evolutionary algorithms have problems to deal with constrained optimization, and it is not different in GA. Some techniques have been developed in an attempt to overcome this difficulty. In most applications of GAs to constrained optimization problems, the penalty function method has been used [11]. In this work was tried an adaptive penalty method which does not require any type of user defined penalty parameter and uses information from the population. Moreover, to make the global search procedure more effective a filter scheme is proposed in which the level of feasibility of the initial population is assured.

4.1.1. Constraints handling

Constraint handling methods used in classical optimization algorithms can be classified into two groups: (i) generic methods that do not exploit the mathematical structure (whether linear or nonlinear) of the constraint, and (ii) specific methods that are only applicable to a special type of constraints. Generic methods, such as the penalty function method, the Lagrange multiplier method, and the complex search method [10, 8] are popular, because each one of them can be easily applied to any problem without much change in the algorithm. But since these methods are generic, the performance of these methods in most cases is not satisfactory.

However, specific methods, such as the cutting plane method, the reduced gradient method, and the gradient projection method [10, 8], are applicable either to problems having convex feasible regions only or to problems having a few variables, as the strategy of choice for constraints handling.

Since GAs are generic search methods, most applications of them to constraint optimization problems have used the penalty function. The penalty function approach involves a number of penalty parameters which must be set right in any problem to obtain feasible solutions. This dependency of GA's performance on penalty parameters has led researchers to devise sophisticated penalty function approaches such as multi-level penalty functions [3], dynamic penalty functions [9], and penalty functions involving temperature-based evolution of penalty parameters with repair operators [22]. All these approaches require extensive experimentation for setting up appropriate parameters needed to define the penalty function [11].

In special situations, closed genetic operators (in the sense that when applied to feasible parents they produce feasible offspring) can be designed if enough domain knowledge is available [13]. Special decoders [16] — that always generate feasible individuals from any given genotype — have been devised, but no applications considering implicit constraints have been published. In this work is considered a chromosome repairing method for the linear constraints presented in Eq. (1) with an adaptive penalty method, introduced for Barbosa and Lemonge [5] in case of infeasible generated offspring.

4.1.2. Chromosome repairing

For this process, initially a randomly population is generated. After this, it's verified the level of violation related to the linear constraints defined in the problem presented in Eq. 1. From this verification it is created two parameters (λ^l, λ^u). The first parameter is called parameter of inferior activation, which refers the individual located in the unfeasible region to activate the constraints by lower bounds, as described in Eq. 4.

$$\mathbf{A}\cdot(\lambda^l \boldsymbol{x}) = \mathrm{b}^l \tag{4}$$

The second parameter is created in a similar way. It is called parameter of superior activation, which refers the individual located in the unfeasible region to activate the constraints by upper bounds, as described in Eq. 5.

$$\mathbf{A}\cdot(\lambda^u \boldsymbol{x}) = \mathrm{b}^u \tag{5}$$

With these activation parameters, a λ, called parameter of feasibility, is randomly generated, between the parameters defined above. These parameters are generated as many times as unfeasible individuals are presented in the initial population. The use of λ guarantees that the imposed constraints will not be violated, but it is not correct to affirm that the individual is completely in feasible region because it might violate the boundaries of design region.

After the application of the parameter of feasibility, the coordinates of the individual are ranked to be corrected if necessary. The coordinates are ranked in a descendant order. In case of any coordinate out of the bounds, redistribution is made to send the individual to the feasible region.

In case of violated upper bound, the excess is relocated in other coordinate as the rank goes on. When more than one coordinate violate the boundaries, the excess is summed to be relocated. When the lower bound is violated, it is taken the complement from the follow coordinates unless it reaches the bound too.

After unfeasible individual be located in the feasible region its coordinates are relocated in the original position of the chromosome. To consider this process is crucial for creating the initial population, since the GA presents difficulty to solve the problem proposed in this work, when it was used without any feasible individual in the initial population. In various tests considered, the optimization process has been finished without any solution found

It is important to mention that the above procedure is used to create the initial population for GA and to create some feasible samples for the global surrogate model used in the global search of the optimization process.

4.1.3. Adaptive Penalty Method (APM)

The adaptive penalty method (APM) scheme presented in Barbosa and Lemonge (2002) [5] adaptively quantifies the penalty coefficients of each constraint using information from the population such as the average of the objective function and the level of violation of each constraint. The fitness function is written as:

$$F(\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & \text{if } \boldsymbol{x} \text{ is feasible,} \\ \overline{f}(\boldsymbol{x}) + \sum_{j=1}^{m} \lambda_j v_j(\boldsymbol{x}) & \text{otherwise} \end{cases} \tag{6}$$

where

$$\overline{f}(\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & \text{if } f(\boldsymbol{x}) > \langle f(\boldsymbol{x}) \rangle, \\ \langle f(\boldsymbol{x}) \rangle & \text{otherwise} \end{cases} \tag{7}$$

and $\langle f(\boldsymbol{x}) \rangle$ is the average of the objective function values in the current population. In Figure 1 feasible as well as infeasible solutions are shown. Among the six infeasible solutions, the individuals #3, #4, #5 and #6 have their objective function values (represented by opened circles), less than the average objective function and, according to the proposed method (see Eq. 7), will have $\overline{f}(\boldsymbol{x})$ given by $\langle f(\boldsymbol{x}) \rangle$. The solutions #1 and #2 have objective function values which are worst than the population average and thus will have $\overline{f}(\boldsymbol{x}) = f(\boldsymbol{x})$.

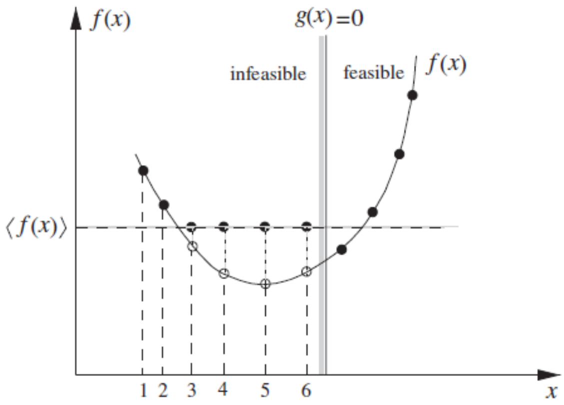

Figure 1. A pictorial description of the function $\bar{f}(\boldsymbol{x})$.

The penalty parameter is defined at each generation by:

$$\lambda_j = \left|\left\langle f(\boldsymbol{x})\right\rangle\right| \frac{\left\langle v_l(\boldsymbol{x})\right\rangle}{\sum_{l=1}^{m}\left[\left\langle v_l(\boldsymbol{x})\right\rangle\right]^2} \tag{8}$$

and $\left\langle v_j(\boldsymbol{x})\right\rangle$ is the violation of the l-th constraint averaged over the current population. The idea is that the values of the penalty coefficients should be distributed in a way that those constraints which are more difficult to be satisfied should have a relatively higher penalty coefficient. Details of the proposed definition can be seen in [13]. In the present work, this methodology was implemented in MATLAB environment to be used with GA

4.2. Local Strategy

The SAO [15, 14] methodology is adopted as local strategy on optimization process. SAO methodology decomposes the original optimization problem into a sequence of optimization subproblems, confined into small subregions of optimization design space. Surrogate functions (low-cost) are created and used by the optimizer. A trust region based method is used to update the design variable space for each subproblem (SAO iteration).

Each subproblem defines a SAO iteration. To update the trust region size for each optimization subproblem we considered the approach described in [14] which takes into consideration the accuracy of surrogate functions against the true functions. The main steps involved in the computations are:

1. Compute the expensive and/or nonsmooth objective function and constraints at the central point in the subregion;
2. Construct surrogate model in the subregion;
3. Optimize within the subregion using the surrogate objective function and constraints;
4. Compute the true objective function and constraints at the optimum identified in step 3;
5. Check for convergence;
6. Move/shrink/expand the subregion according to the accuracy of the approximated model compared to the true function and constraint values;
7. Impose local consistency;
8. Check for overall optimization convergence. If it is achieved stop the SAO procedure; otherwise return to step 3.

5. Examples

Firstly some results of the studies with APM are presented, considering only GA. These tests were considered to check if APM works properly when it coupled with the GA from MATLAB global optimization toolbox. The problem considered was the optimization of a welded beam design.

The objective in this problem is to minimize the cost $C(h, l, t, b)$ of the beam where $h \in [0.125, 10]$, and $0.1 \leq l, t, b \leq 10$, subject to constraints on shear stress (τ_{MAX}), bending stress in the beam (σ_{MAX}), buckling load on the bar (Pc), end deflection of the beam (δ_{MAX}), and side constraints. In Figure 2 the welded beam assembly is shown with its main variables.

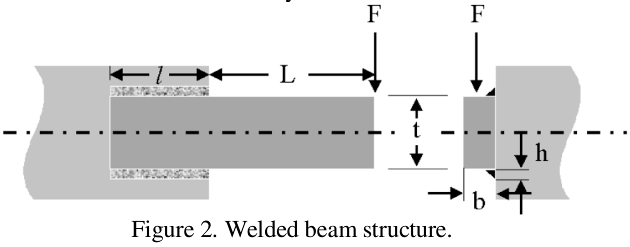

Figure 2. Welded beam structure.

The problem can be formulated as follows:

$$\begin{aligned} &\text{Minimize: } C(h,l,t,b) = 1.10471h^2l + 0.04811tb(14+l) \\ &\text{subject to: } g_1(\tau) = \tau(h,l) - 13600 \le 0 \\ &\qquad g_2(\sigma) = \sigma(t,b) - 30000 \le 0 \\ &\qquad g_3(b,h) = h - b \le 0 \\ &\qquad g_4(Pc) = 6000 - Pc(t,b) \le 0 \\ &\qquad g_5(\delta) = \delta(t,b) - 0.25 \le 0 \end{aligned} \tag{9}$$

where

$$\begin{aligned} &\tau(h,l) = \sqrt{(\tau')^2 + (\tau'')^2 + {l\tau'\tau''}/{\varphi}}; \quad \tau' = \frac{6000}{\sqrt{2}hl}; \\ &\tau'' = \frac{6000(14+0.5l)\varphi}{2(0.707hl(l^2/12 + 0.25(h+t)^2))} \\ &\varphi = \sqrt{0.25(l^2 + (h+t)^2)}; \quad \sigma(\mathrm{x}) = \frac{504000}{t^2b}; \\ &\delta(\mathrm{x}) = \frac{21952}{t^3b}; \quad Pc(\mathrm{x}) = 64746.022(1 - 0.0282346t)tb^3 \end{aligned} \tag{10}$$

To test the implemented methodology, the results found were compared with results obtained using the tools from global optimization toolbox from MATLAB. It uses the Augmented Lagrangian GA (ALGA) to solve nonlinear constraints problems without integer constraints. For statistics calculation a total of 50 runs were performed.

Table 1 shows a comparison between the obtained results where the first row represent the results obtained using APM methodology with MATLAB's GA, and the second one the results obtained using ALGA from MATLAB. The best value found is presented in the reference Bernardino et al. (2007) [5] and corresponds to a final cost equal to 2.38122.

Table 1. Results obtained in GA's test with APM and ALGA

Cons. handling method	*Best*	*Average*	*Std. Dev*	*Worst*	*Function eval. (for best)*
APM	2,45628	4,54617	0,99661	6,86426	76800
ALGA	2,47504	4,46035	1,39126	7,96005	102200

As can be seen, the results presented by APM were better than the ones presented by ALGA, considering the accuracy with the reference and the number of function evaluation.

Now it is presented the proposed hybrid combinations applied for a case study presented in [6]. The characteristics of the problem are quite simple, as the main application purpose was to study and evaluate the basic optimization problem behavior under simplified conditions easier to control. With simple geometry, the problem has as design variables the wells rates and the time that the control cycles will be exchanged. The Eqs 2 and 3 describe these variables.

The reservoir studied in this problem has the characteristics shown on Table 2. In Figure 3 can be seen the wells arrangement (injection and production wells) and the regions separated according to the horizontal permeability in the reservoir. The horizontal permeability (k_h) in the injection well (I-1) region is 1000 mD, the kh near the production well P-1 is 500 mD while near the production well P-2 is 1500 mD.

Table 2. Characteristics from the reservoir considered.

Simulation mesh	51 (510 m) x 51 (510 m) x 1 (4 m)
Porosity	30%
Vertical permeability vertical (k_v)	10% de k_h
Rock compressibility at 200 kgf/cm²	5 . 10^{-5} (kgf/cm²)$^{-1}$
Contact between fluids	Without contact WOC and GOC
Pressure of saturation (P_{sat})	273 kgf/cm²
Viscosity at T_{res}, P_{sat}	0,97 cp
Gas-Oil formation ratio (GOFR)	115,5 m³/m³ std

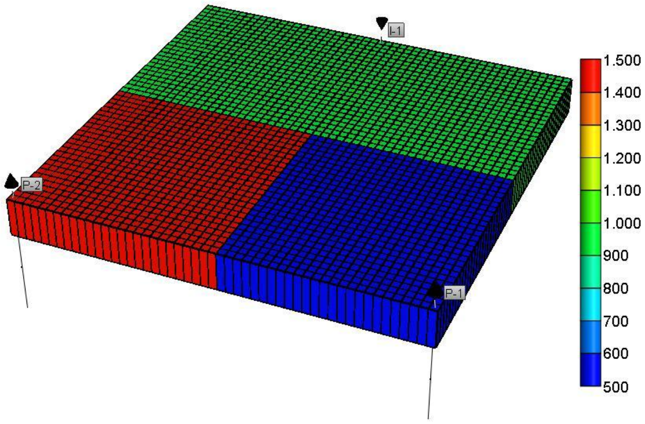

Figure 3. Permeability regions.

For this problem, $Q_{prod,\max}$ may not exceed 40 m³/day, and the individual flow cannot exceed 30 m³/day. The $Q_{inj,\max}$ may not exceed 44 m³/day. Regarding the concession period, the total period considered is 16 years.

To build the global surrogates a total of $5n_{var}$ samplings are selected over the investigated domain. When using SAO strategy, local surrogates are built considering $(2n_{var} + 1)$ samplings as suggested in [6]. The global, local and hybrid strategies described in this work are used to solve this particular example.

In this problem, 2 cases are considered adopting different management strategies. The cases studied are:

Case 1 – Non-full capacity operation with fixed time to execute the control cycles; this case considers 3 control cycles only but with the consideration of the non-full capacity operation the complexity of the problem is greatly increased as commented before (by the consideration of a great number of constraints);

Case 2 – Non-full capacity operation with variable time to execute the control cycles; this case considers 3 control cycles, as the previous case, and also considers the time definition for the control cycles as design variables. This is the most complex case because of the consideration of the constraints and the multimodality imposed by variables of time. For each case the number of control cycles and design variables involved are summarized on Table 3.

Table 3. Cases studied

Case	*Control cycles*	*Variables*
1	3	9 (6 q_{prod} + 3 q_{inj})
2	3	11 (6 q_{prod} + 3 q_{inj} + 2 t)

Tables 4 and 5 are shown the results obtained by the hybrid strategy considering only GA with APM, and the reference results obtained by SAO technique is also shown.

Table 4. Obtained results in case 1.

	$f(\boldsymbol{x})$ (10^6 U. M.)	*function evaluations*
SAO	1,72299	421
Hybrid	1,72267	441

Table 5. Obtained results in case 2.

	$f(\boldsymbol{x})$ (10^6 U. M.)	*function evaluations*
SAO	1,73314	673
Hybrid	1,72921	291

According with the above results, it can be seen that the solutions obtained by hybrid strategies were satisfactory. The results obtained in case 2 presents a low discrepancy in relation to the reference value but the number of function evaluation is less than the half of the one required by SAO.

Other studies on problems involving reservoir with characteristics similar to real situations, aiming to verify the tool proposed for solve such problems are current being analyzed.

6. Conclusion

Waterflooding management problem was conducted in this work by a hybrid strategy considering surrogate models. These methodologies appear to be a good way to deal with this kind of problem, when considering the characteristic of the function involved on this problem formulation. The cases considered involve a high nonlinear objective function and also presents several constraints functions. These two aspects together turn out the problem too difficult to be solved by the EA's, in charge of the first stage of the optimization. To improve the performance of the GA considered in global search process, some tools were considered to increase the capability to find a good feasible solution.

As can be seen in the results, hybrid converged to reported reference solution with fewer function simulation runs. The local search executed on hybrid methodology depends on the initial point, which is given by the global search; if a bad solution is given as initial point to the local search algorithm hardly a good solution will be found.

It is important to emphasize that the solution provided by the reported reference (SAO algorithm) is a result found after many tries in order to find a good solution, as typically when local search algorithms are used. The hybrid methodologies considered in this work overcome then need to perform several initial points tries on reservoir optimization problems.

7. Acknowledgment

The authors acknowledge the financial support for this research given by PRH-26, from ANP (National Petroleum Agency – Brazil), CNPq (National Research Council – Brazil), FACEPE, PETROBRAS, and UFPE.

8. References

1. A A Giunta, Use of Data Sampling, Surrogate Models, and Numerical Optimization in Engineering Design, Paper AIAA-2002-0538 in Proceedings of the 40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 2002.
2. A Forrester, A Sobester, A Keane, Engineering Design Via Surrogate Modelling: A Practical Guide, Chichester: Wiley, 228 p. ISBN 0470060689, 2008.
3. A Homaifar, S H-V Lai, X Qi, Constrained optimization via Genetic Algorithms, Simulation 62/4, 242–254, 1994.
4. B Horowitz, S M B Afonso, C V P Mendonça, Rate Control Optimization of Waterflooding Management, EngOpt – 2nd International Conference on Engineering Optimization, Lisbon-Portugal, 2010.
5. A C C Lemonge, H J C Barbosa, C C H Borgesc, F B S Silva, Constrained Optimization Problems In Mechanical Engineering Design Using A Real-Coded Steady-State Genetic Algorithm, In IX Congreso Argentino de Mecánica Computacional / II Congreso Sudamericano de Mecánica Computacional / XXXI CILAMCE – Congreso Ibero-Latino-Americano de Mátodos Computacionales en la Inginería, 2010.
6. B Horowitz, S M B Afonso, C V P Mendonça, R B Wilmersdorf, Rate Control Optimization of Waterflooding Management, In XXX CILAMCE – Congresso Ibero Lantino Americano de Métodos Computacionais em Engenharia, 2009.
7. D E Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.
8. G V Reklaitis, A Ravindran, K M Ragsdell, Engineering Optimization Methodsand Applications,Wiley, New York, 1983.
9. J A Joines, C R Houck, On the Use of Nonstationary Penalty Functions to Solve Nonlinear Constrained Optimization Problems with GAs, in: Z. Michalewicz, ed., Proceedings of the International Conference on Evolutionary Computation – IEEE Press, Piscataway, 579–584, 1994.
10. K Deb, Optimization for Engineering Design: Algorithms and Examples, Prentice-Hall, New Delhi, 1995.
11. K Deb, An Efficient Constraints Handling Method for Genetic Algorithms, Preprint submitted to Elsevier Prepoint, 1998.
12. M Gen, R Cheng, Genetic Algorithm and Engineering Optimization, John Wiley & Sons, Inc., 2000.
13. M Schoenauer, Z Michalewicz. Evolutionary Computation at the Edge of Feasibility, In Parallel Problem Solving from Nature—PPSN IV, Voigt H-M, Ebeling W, Rechenberg I, Schwefel H-P (eds), Lecture Notes in Computer Sciences, Vol. 1141, Springer: Berlin, 245–254, 1996.
14. M S Eldred, A A Giunta, S S Collis, Second-Order Corrections for Surrogate-Based Optimization with Model Hierarchies, Paper AIAA-2004-4457 in Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, 2004.
15. N Alexandrov, J E Dennis Jr, R M Lewisand, V Torezon, A Trust Region Framework for Managing the Use of Approximation Models in Optimization, NASA/CR-201745; ICASE Report No. 97-50, 1997.
16. S Koziel, Z Michalewicz, Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization, Evolutionary Computation, 7(1):19–44, 1999.
17. S M B Afonso, B Horowitz, R B Willmersdorf, Comparative Study of Surrogate Models For Engineering Problems In 7th ASMO_UK/ISSMO International Conference on Engineering Design Optimization, Bath-UK, 2008.
18. S R Otto, J P Denier, An Introduction to Programming and Numerical Methods in MATLAB. Springer-Verlag London Limited, 2005.
19. T W Simpson, T M Mauery, J K Korte, F Mistree, Kriging Models for Global Approximations in simulation-Based Multidisciplinary Design Optimization, AIAA Journal, 39(12):2233-2241, 2001.
20. V J Romero, J V Burkardt, M D Gunzburger, J S Peterson, Comparison of pure 'Latinized' Centroidal Voronoi tessellation against various other statistical sampling methods, Reliability Engineering and System Safety 91, 1266-1280, 2006.
21. W H Press, S A Teukolsky, W T Vetterling, B P Flannery, Numerical recipes in Fortran: the art of scientific computing. 2nd ed., Cambridge, Cambridge University Press, 1992.
22. Z Michalewicz, N Attia, Evolutionary Optimization of Constrained Problems. in: A V Sebald and L J Fogel, eds., Proceedings of the Third Annual Conference on Evolutionary Programming, World Scientific, Singapore, 98–108, 1994.
23. Z Michalewicz, Genetic Algorithm + Data Structure = Evolution Programs, 3rd ed., Springer-Verlag, New York, 1996.
24. Z Michalewicz, D Dasgupta, editors, Evolutionary Algorithms in Engineering Applications, Springer-Verlag, 1997.